

Chương 3: JQuery

Giảng Viên: ThS. Vũ Minh Sang

Nội dung

Giới thiệu jQuery và cách sử dụng

Selector trong jQuery

Jquery HTML

Attributes trong jQuery

Event trong jQuery

Effect trong jQuery

Manipulation trong jQuery

Giới thiệu jQuery và cách sử dụng



| jQuery là gì?

- **jQuery:** Thư viện của Javascript giúp đơn giản hóa cách viết Javascript và tăng tốc độ xử lý các sự kiện trên trang web, hỗ trợ các nhà lập trình web tạo ra các tương tác trên website một cách nhanh nhất
- **jQuery** được khởi xướng bởi **John Resig** (hiện là trưởng dự án của **Mozilla**) vào năm 2006, **jQuery** có mã nguồn mở và hoàn toàn miễn phí
- **jQuery** có cấu trúc rất mạch lạc và theo hệ thống. Cách viết code của **jQuery** được vay mượn từ các nguồn mà các **web designer** đa phần đã biết như HTML và CSS





| jQuery có thể làm được gì?

- Là một thư viện **Javascript** miễn phí
- Hỗ trợ lập trình viên viết **Javascript** nhanh hơn, dễ dàng, ngắn gọn và kết quả thực thi tốt hơn.
- Đơn giản hóa việc quản lý và xử lý các thành phần trong **HTML/DOM**
- Thay đổi giao diện của một trang web trong việc sử dụng **CSS**
- Tương tác với người dùng qua **Event Handler API**
- Tạo **hiệu ứng động** cho trang web
- Đơn giản hóa **AJAX**.
- Thay đổi cách viết **Javascript**





Tại sao chọn jQuery

- Miễn phí hoàn toàn
- Một lượng lớn cộng đồng hỗ trợ plug in
- Xoá nhòa sự khác biệt giữa trình duyệt
- Tương thích với nhiều trình duyệt.
- Cách sử dụng đơn giản.
- Dung lượng của JQuery chưa tới 100KB
- Xử lý có nhanh và khả năng mở rộng.



Sử dụng jQuery

- jQuery là một thư viện Javascript do vậy để sử dụng phải chèn nó vào trang web như một file javascript thông thường.

```
<script src="duong_dan_toi_file_jquery"></script>
```

- duong_dan_toi_file_jquery: có thể tải trên các server của google.com, jquery.com.
 - Hoặc sử dụng jquery cdn để thêm trực tiếp link vào trang web
- Thứ tự trong các file trong một trang html

```
<link rel="stylesheet" href="stylesheet.css"/>
```

```
<script src="javascript.js"></script>
```

```
<script src="first-jquery.js"></script>
```

Thời điểm thực thi

- Xét 2 sự kiện **onload** trên **javascript** và **ready()** trên **jQuery** (2 sự kiện có tác vụ tương tự nhau, tuy nhiên thời điểm thực thi khác nhau)
 - **onload** trên **javascript** được thực thi khi toàn bộ nội dung trang web (**document**) được tải hết về browser (trình duyệt)
 - **ready()** trên **jQuery** được thực thi khi **DOM đã sẵn sàng** (cấu trúc trang web đã được tải)

Selector trong jQuery

Selector trong jQuery

- Dùng để chọn các phần tử trong trang web html theo ý muốn
- Cú pháp: `$('selector').action()`
 - **\$**: xác định jQuery
 - **selector**: truy vấn với thành phần html
 - **action**: thể hiện hành động trên thành phần selector được chọn
 - **VD**: `$("p").hide();`

Selector trong jQuery

- Cú pháp: `$('selector').action(function() {})`

- Xử lý bằng các hàm tự định nghĩa;

VD:

```
$("input[type='button']").click(function() {  
    $("p").hide();  
});
```

- Các loại selector
 - Basic
 - Attribute
 - Form
 - jQuery Extensions

Selector - Basic



Selector	Ý nghĩa
All Selector ("*")	Chọn tất cả các thành phần trong trang web Ví dụ: \$("*") hoặc \$(document):
Element Selector ("element")	Chọn tất cả các thành phần dựa trên tên thẻ . Ví dụ: \$("p") hoặc \$("h1") hoặc \$("a")
ID Selector ("#id")	Chỉ chọn một thành phần có ID giống tên ID kèm theo. Ví dụ: \$("#test_id")
Class Selector (".class")	Chọn tất cả các thành phần có class giống tên class kèm theo. Ví dụ: \$(".test")
Element, Class Selector ("element.class")	Chọn tất cả các thành phần có element có class giống tên class kèm theo. Ví dụ: \$("div.test")
Multiple Selector ("selector1, selector2, selectorN")	Chọn tất cả các thành phần được chỉ định. Ví dụ: \$(".class1, .class2, .class3")

```
<ul id="nav">
```

```
<li><a href="#">Home Page</a></li>
```

```
<li><a href="#">About Me</a></li>
```

```
<li><a href="#">Forum</a></li>
```

```
<li><a href="ebook.pdf">Ebooks</a></li>
```

```
<ul>
```

```
<li><a href="#">jQuery</a></li>
```

```
<li><a href="#">CSS</a></li>
```

```
<li><a href="#">HTML</a></li>
```

```
<li><a href="#">C#</a></li>
```

```
</ul>
```

```
<li><a href="http://www.jquery.com">Tutorials</a></li>
```

```
<li><a href="#">Photoshop</a></li>
```

```
<ul>
```

```
<li><a href="#">Action</a></li>
```

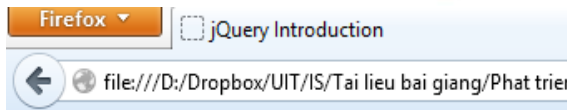
```
<li><a href="#">Effect</a></li>
```

```
<li><a href="#">Plugins</a></li>
```

```
</ul>
```

```
<li><a href="mailto:email@yahoo.com">Email</a></li>
```

```
</ul>
```



jQuery Introduction

- [Home Page](#)
- [About Me](#)
- [Forum](#)
- [Ebooks](#)
 - [jQuery](#)
 - [CSS](#)
 - [HTML](#)
 - [C#](#)
- [Tutorials](#)
- [Photoshop](#)
 - [Action](#)
 - [Effect](#)
 - [Plugins](#)
- [Email](#)

Selector - Basic

- Thêm vào file .css

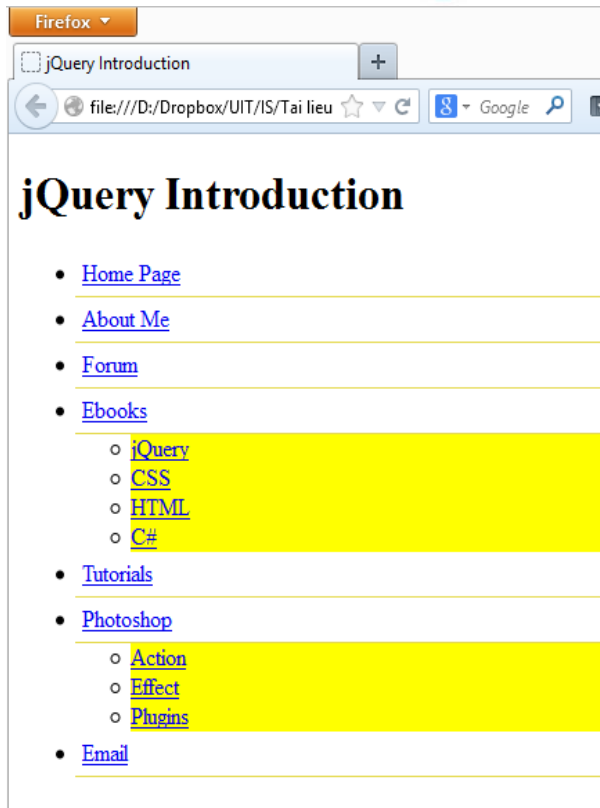
```
.highlight {  
border-bottom: 1px solid #e6db55;  
padding: 5px;}
```

- Chỉ định dạng các phần (cấp 1) như: Home Page, About, Forum...

```
$(document).ready(function() {  
    $('#nav > li').addClass('highlight');  
});
```

- Chỉ định dạng các phần (cấp 2) như: jQuery, CSS, Action...

```
$(document).ready(function() {  
    $('#nav > li').addClass('highlight');  
    $('#nav li:not(.highlight)').addClass('background');  
});
```



Selector - Attribute



Selector	Ý nghĩa
Has Attribute Selector [name]	Chọn các thành có thuộc tính là "name" với giá trị bất kì. Ví dụ <code>\$('input[name]')</code> Demo
[attribute=value]	Chọn các thành phần có các thuộc tính với giá trị là "value". Ví dụ <code>\$('input[name="text"]')</code> Demo1 , Demo2
[attribute!=value]	Chọn các thành phần có thuộc tính khác với giá trị "value". Ví dụ <code>\$('input[name!="text"]')</code> Demo
[attribute*=value]	Chọn các thành phần có thuộc tính với giá trị có chứa chuỗi "value". Ví dụ <code>\$('input[name*="text"]')</code> Demo
[attribute^="value"]	Chọn các thành phần có thuộc tính với giá trị bắt đầu có chứa value. Ví dụ <code>\$('input[name^="text"]')</code> Demo
[attribute\$=value]	Chọn các thành phần có thuộc tính với giá trị kết thúc có chứa chuỗi "value". Ví dụ <code>\$('input[name\$="text"]')</code> Demo
[attribute~=value]	Chọn các thành phần có thuộc tính với giá trị có chứa từ "value". Ví dụ <code>\$('input[name~="text"]')</code> Demo



Selector - Form

Selector	Ý nghĩa
:button	Chọn tất cả các phần tử button và các phần tử có thuộc tính type= "button" Demo
:checkbox	Chọn tất cả các phần tử có type= "checkbox" Demo
:checked	Chọn tất cả các phần tử mà đã được check rồi Demo
:disabled	Chọn tất cả các phần tử đó bị disable Demo
:enabled	Chọn tất cả các phần tử mà đã được kích hoạt Demo
:focus	Chọn phần tử nếu nó hiện đang tập nhận Focus Demo
:password	Chọn tất cả các phần tử có kiểu là password Demo
:radio	Chọn tất cả các phần tử có kiểu là radio Demo
:selected	Chọn tất cả các phần tử option của select đã được chọn Demo
:reset	Chọn tất cả các phần tử có kiểu là reset Demo
:text	Chọn tất cả các phần tử có kiểu là text Demo
:submit	Chọn tất cả các phần tử có kiểu là submit Demo

Selector - Extensions



Selector	Ý nghĩa
:eq(index)	Chọn phần tử tại chỉ số n trong các thiết lập phù hợp Demo
:even	Chọn các phần tử chẵn, đánh dấu chỉ mục từ số 0 Demo
:odd	Chọn các phần tử lẻ, đánh dấu chỉ mục từ số 0 Demo
:first	Chọn các thành phần phù hợp đầu tiên (demo bên dưới)
:last	Chọn thành phần phù hợp cuối cùng (demo bên dưới)
:gt(index)	Chọn tất cả các thành phần ở một chỉ số lớn hơn chỉ số trong các thiết lập phù hợp Demo
:header	Chọn tất cả các thành phần mà nó là tiêu đề, như h1..h6 Demo
:hidden	Chọn tất cả các thành phần đó được ẩn Demo
:file	Chọn tất cả các phần tử của loại là "type = file" Demo1
:lt(index)	Chọn tất cả các thành phần tại một chỉ số nhỏ hơn chỉ số trong các thiết lập phù hợp Demo

Selector - Extensions



Selector	Ý nghĩa
<code>\$(this)</code>	Chọn thành phần hiện tại tại Demo
<code>\$("p.intro")</code>	Chọn tất cả các tag <code><p></code> có <code>class="intro"</code> Demo
<code>\$("p:first")</code>	Chọn tag <code><p></code> đầu tiên Demo
<code>\$("ul li:first")</code>	Chọn tag <code></code> đầu tiên trong tag <code></code> đầu tiên Demo
<code>\$("ul li:first-child")</code>	Chọn tag <code></code> đầu tiên của tất cả các tag <code></code> Demo
<code>\$("[href]")</code>	Chọn tất cả các thành phần có thuộc tính <code>href</code> Demo
<code>\$("p:last")</code>	Chọn tag <code><p></code> cuối Demo
<code>\$("ul li:last")</code>	Chọn tag <code></code> cuối cùng trong tag <code></code> cuối cùng Demo
<code>\$("ul li:last-child")</code>	Chọn tag <code></code> Cuối cùng trong tất cả các tag <code></code> Demo

jQuery HTML

| jQuery HTML



jQuery	Ý nghĩa
jQuery Get	Lấy giá trị từ các đối tượng
jQuery Set	Đặt dữ liệu lên các đối tượng
jQuery Add	Thêm thành phần vào đối tượng
jQuery Remove	Loại bỏ một thành phần trong đối tượng
jQuery CSS Class	Các thành phần class CSS
jQuery CSS	Các thành phần CSS
jQuery Dimensions	Kích thước các thành phần HTML

jQuery HTML – Get, Set

Hàm	Ý nghĩa
<code>text()</code>	Đặt, lấy một nội dung text hoặc một thành phần được chọn
<code>html()</code>	Đặt, lấy một nội dung text hoặc một thành phần được chọn (hoặc một tag đánh dấu văn bản)
<code>val()</code>	Đặt, lấy một giá trị từ các thành phần trên Form

Demo 1

| jQuery HTML – Add



Hàm	Ý nghĩa
append ()	Thêm một nội dung vào cuối thành phần được chọn: Demo
prepend ()	Thêm một nội dung vào đầu thành phần được chọn: Demo
after ()	Thêm một nội dung vào phía sau thành phần đang được chọn: Demo
before ()	Thêm một nội dung vào phía đầu thành phần đang được chọn: Demo



| jQuery HTML – Remove



Hàm	Ý nghĩa
remove ()	Xóa thành phần được chọn kể cả các thành phần con của nó Demo
empty ()	Xóa nội dung của thành phần được chọn Demo

jQuery HTML – CSS Classes

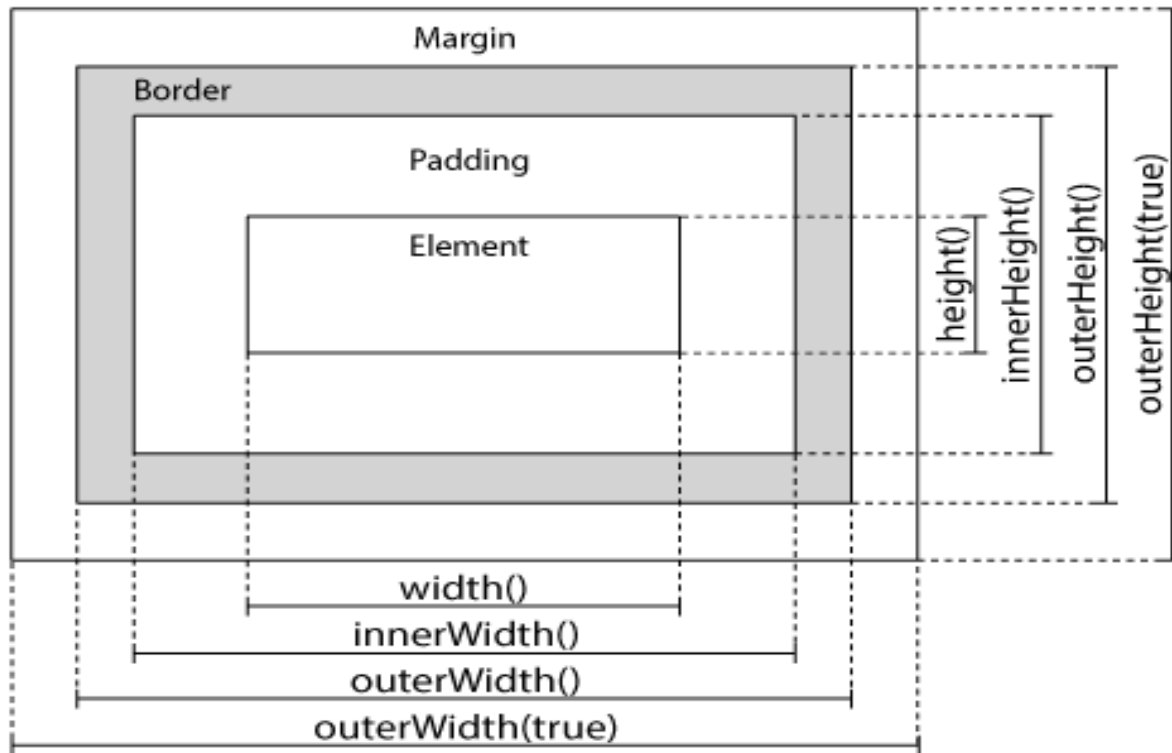


Hàm	Ý nghĩa
<code>addClass()</code>	Thêm một hoặc nhiều class vào những thành phần được chọn Demo1 , Demo2
<code>removeClass()</code>	Xóa một hoặc nhiều class từ những thành phần được chọn Demo
<code>toggleClass()</code>	Thêm/xóa những class từ những thành phần được chọn Demo
<code>css()</code>	Đặt hoặc lấy một loại thuộc tính (slide kế)

jQuery HTML – CSS Classes – CSS

Hàm	Ý nghĩa
Lấy css	<code>\$("p:last").css("background-color")</code> Demo
Đặt css	<code>\$("p").css("background-color","yellow")</code> Demo
Đặt nhiều css	<code>\$("p").css({"background-color":"yellow","font-size":"200%"});</code> Demo

jQuery Dimensions



jQuery Dimensions



Hàm	Ý nghĩa
<code>width()</code>	Chiều rộng của đối tượng
<code>height()</code>	Chiều cao của đối tượng Demo1 , Demo 2
<code>innerWidth()</code>	Chiều rộng của vùng inner
<code>innerHeight()</code>	Chiều cao của vùng inner Demo
<code>outerWidth()</code>	Chiều rộng của vùng outer
<code>outerHeight()</code>	Chiều cao của vùng outer Demo

Attributes trong jQuery



Attributes

- Cung cấp các phương thức giúp lập trình viên có thể tương tác với các thuộc tính của các phần tử trong DOM



Attributes



Attributes	Ý Nghĩa
<code>.addClass()</code>	Thêm một CSS vào cho mỗi tập phần tử phù hợp. Demo
<code>.attr()</code>	Lấy giá trị của một thuộc tính của một phần tử Demo
<code>.hasClass()</code>	Xác định xem bất kỳ các phần tử phù hợp được phân công lớp cho trước. Demo
<code>.prop()</code>	Lấy hoặc đặt giá trị thuộc tính của phần tử được chọn Demo

Attributes



Attributes	Ý nghĩa
<code>.removeAttr()</code>	Hủy bỏ một thuộc tính từ mỗi phần tử trong tập hợp các phần tử phù hợp. Demo
<code>.removeClass()</code>	Hủy bỏ một hoặc nhiều CSS class của phần tử được chỉ định. Demo
<code>.removeProp()</code>	Hủy 1 thuộc tính cho tập hợp các phần tử phù hợp.
<code>.toggleClass()</code>	Thêm hoặc loại bỏ một hoặc nhiều lớp từ mỗi phần tử trong tập hợp các phần tử phù hợp, tùy thuộc vào sự hiện diện của lớp hoặc giá trị của các đối số chuyển đổi. Demo
<code>.val()</code>	Lấy giá trị của phần tử Demo

Event trong jQuery

Events

- Cú pháp:

```
$(selector).event(function() {  
    //Code here  
});
```

- Các loại event trong jquery:

- Document Loading
- Form Events
- Browser Events
- Mouse Events
- Event Handler Attachment
- Event Object
- Keyboard Events

Events – Document loading

Events	Ý Nghĩa
<code>.load()</code>	xảy ra khi document và các phần tử con của nó được load xong. Demo
<code>.ready()</code>	xảy ra khi một tài liệu DOM đã được load hoàn toàn. Demo
<code>.unload()</code>	xảy ra khi một user đi qua một trang đang load, trang chưa kịp load xong thì đã chuyển hướng đi trang khác, trang được load lại hoặc trình duyệt đóng lại. Demo

Form Events



Events	Ý Nghĩa
.blur()	xảy ra khi một phần tử bị mất focus. Demo
.change()	xảy ra khi một phần tử bị thay đổi. Demo
.focus()	xảy ra khi phần tử được focus. Demo
.select()	xảy ra khi text được chọn trong text area hoặc phần tử input text . Demo
.submit()	xảy ra khi form được submit. Demo

Browser Events



Events	Ý Nghĩa
<code>.scroll()</code>	Xảy ra khi người dùng cuộn thanh cuộn của một phần tử. Demo
<code>.resize()</code>	Xảy ra khi user điều chỉnh kích thước của một phần tử. Demo



Mouse Events



Events	Ý Nghĩa
<code>.click()</code>	Xảy ra khi click chuột. Demo
<code>.dblclick()</code>	Xảy ra khi double click chuột. Demo
<code>.hover()</code>	Xảy ra khi rê chuột vào hoặc ra khỏi 1 phần tử. Demo
<code>.mousedown()</code>	Xảy ra khi click chuột vào phần tử và không cần nhả click. Demo
<code>.mouseenter()</code>	Xảy ra khi trỏ chuột ở trên một phần tử. Demo

Mouse Events



Events	Ý Nghĩa
<code>.mouseleave ()</code>	Xảy ra khi trỏ chuột rời khỏi 1 phần tử
<code>.mousemove ()</code>	Xảy ra mỗi khi trỏ chuột di chuyển bên trong 1 phần tử. Demo
<code>.mouseout ()</code>	Xảy ra khi trỏ chuột di chuyển khỏi phần tử
<code>.mouseover ()</code>	Xảy ra khi trỏ chuột ở trên một phần tử
<code>.mouseup ()</code>	Xảy ra khi trỏ chuột nhả ra khỏi phần tử. Demo

Events - Attachment



Events	Ý nghĩa
.bind()	Đính kèm một hay nhiều sự kiện cho phần tử. Demo1 , Demo2
.delegate()	Đính kèm một hay nhiều sự kiện cho các phần tử là con của phần tử được chọn Demo1 , Demo2
.trigger()	Tạo ra một sự kiện cho phần tử được chọn. Demo

Events - Attachment



Events	Ý nghĩa
<code>.triggerHandler()</code>	Giống như <code>.trigger()</code> nhưng không gây ra trạng thái mặc định của sự kiện đang xảy ra. Demo
<code>.unbind()</code>	Loại bỏ các sự kiện khỏi các phần tử được chọn. Demo
<code>.undelegate()</code>	Loại bỏ các sự kiện được thêm vào bởi <code>delegate()</code> . Demo

Events – Event Object



Events	Ý Nghĩa
<code>event.preventDefault()</code>	Dừng ngay một sự kiện đang xảy ra. Demo
<code>event.timeStamp</code>	Thuộc tính chứa thời gian tính bằng miligiay khi sự kiện xảy ra. Demo
<code>event.pageX</code> <code>event.pageY</code>	Hiển thị tọa độ x , y của trỏ chuột. + Demo x + Demo y

Keyboard Events



Events	Ý Nghĩa
.focusin()	Xảy ra khi một phần tử được focus bằng trò chuột và kể cả bàn phím. Ví dụ: tab Demo
.focusout()	Xảy ra khi phần tử bị mất focus . Demo

Effect trong jQuery



Effects trong JQuery

- Các loại hiệu ứng thường sử dụng được JQuery hỗ trợ:
 - Fading (hiệu ứng làm mờ)
 - Custom (hiệu ứng tùy chỉnh)
 - Basics
 - Sliding



Effects - Fading



Effects	Ý Nghĩa
.fadeIn ()	Làm đậm phần tử sau khi nó được làm mờ cho đến khi opacity =1 (độ sáng) Demo
.fadeOut ()	Làm mờ phần tử được chọn cho đến khi opacity =0 (độ sáng) Demo
.fadeTo ()	Sự kết hợp của 2 phương thức trên. FadeTo cho phép bạn làm mờ hoặc làm đậm phần tử với opacity (độ sáng) được chỉ định. Demo

Effects - Custom



Effects	Ý Nghĩa
<code>.hide()</code>	Ẩn phần tử. Demo
<code>.show()</code>	Hiện phần tử. Demo
<code>.toggle()</code>	Nếu phần tử đang hiện thì sẽ chuyển sang ẩn và ngược lại. Demo

Effects - Basic



Effects	Ý Nghĩa
.animate()	Phương thức tạo ra các chuyển động về hình dạng, kích thước, vị trí, ... cho phần tử. Demo
.delay()	Tạo giá trị trễ cho phần tử khi thực hiện các hàm Demo
.stop()	Tắt các hiệu ứng đang chạy Demo

Effects - Sliding



Effects	Ý Nghĩa
.slideDown()	Đưa phần tử trở lại kích thước ban đầu. Demo
.slideUp()	Thu phần tử nhỏ lại cho đến khi không nhìn thấy. Demo
.slideToggle()	Đưa phần tử về kích thước ngược lại với kích thước hiện tại. Đây là hàm kết hợp giữa SlideUp và SlideDown. Demo

Manipulation trong jQuery



Effects trong JQuery

- Các phương thức cơ bản thao tác trên DOM:
 - Style Properties (thêm thuộc tính)
 - DOM Replacement (thay thế)
 - DOM Removal (xóa)
 - DOM Insertion (chèn)
 - Around
 - Inside
 - Outside



Style Properties



Phương Thức	Ý Nghĩa
<code>.css()</code>	Css cho một phần tử. Demo
<code>.width()</code>	Thiết lập hoặc trả về chiều rộng của phần tử. Demo
<code>.height()</code>	Thiết lập hoặc trả về chiều cao của phần tử. Demo
<code>.scrollLeft()</code>	Thiết lập hoặc trả về tung độ của scrollbar cho phần tử được chọn. Demo
<code>.offset()</code>	Trả về vị trí (tương đối so với Document) của phần tử được chọn đầu tiên Demo
<code>.position()</code>	Trả về vị trí (tương đối so với phần tử cha) của phần tử được chọn đầu tiên Demo

DOM Replacement



Phương Thức	Ý Nghĩa
<code>.replaceAll()</code>	Thay thế nội dung của phần tử được chọn với nội dung mới. Demo
<code>.replaceWith()</code>	Giống <code>replaceAll()</code> . Demo

DOM Removal



Phương Thức	Ý Nghĩa
.detach ()	Loại bỏ phần tử được chọn(nhưng giữ một bản copy) . Demo
.empty ()	Loại bỏ tất cả phần tử con và nội dung trong phần tử được chọn. Demo
.remove ()	Loại bỏ phần tử được chọn. Demo
.unwrap ()	Loại bỏ phần tử cha của phần tử được chọn. Demo

DOM Insertion Around



Phương Thức	Ý Nghĩa
.wrap()	Bao mỗi phần tử được chọn bằng phần tử HTML.
.wrapAll()	Bao các phần tử được chọn bằng một phần tử HTML. Demo
.wrapInner()	Bao nội dung của mỗi phần tử được chọn bằng một phần tử HTML. Demo

DOM Insertion Inside



Phương Thức

Ý Nghĩa

.append()

Cho phép chúng ta thêm vào nội dung bên trong thành phần tìm được, nhưng sau phần tử con cuối cùng của thành phần đó. [Demo](#)

.appendTo()

Giống .append() và có thể di chuyển một thành phần đến một vị trí khác. [Demo](#)

.prepend()

Cho phép chúng ta thêm vào nội dung bên trong thành phần tìm được, nhưng trước các phần tử con của thành phần đó. [Demo](#)

.prependTo()

Giống .prepend() và có thể di chuyển một thành phần đến một vị trí khác. [Demo](#)

.html()

1. Lấy toàn bộ nội dung bao gồm cả các thẻ HTML.
2. Khi nhận vào giá trị, nó sẽ thay thế nội dung của thành phần bằng một nội dung mới. [Demo](#)

.text()

Có 2 tác dụng tương tự .html(), chỉ khác là phương thức .text() sẽ lấy hoặc thêm vào các giá trị text (không bao gồm các thẻ HTML).
[Demo](#)

DOM Insertion Outside



Phương Thức	Ý Nghĩa
.after()	Cho phép chúng ta thêm vào nội dung, hoặc các thành phần HTML vào ngay sau phần tử tìm được trong cấu trúc HTML. Demo
.before()	Ngược lại với .after(), before() sẽ thêm vào nội dung phía trước phần tử tìm được. Demo
.insertAfter()	Chèn tài liệu HTML hoặc phần tử khác vào phía sau của phần tử được chọn.
.insertBefore()	Chèn tài liệu HTML hoặc phần tử khác vào phía trước của phần tử được chọn.



Cảm ơn đã theo dõi

Hy vọng cùng nhau đi đến thành công.